REMARKS

Applicants have received the Office action dated December 30, 2009, in which the Examiner rejected claims 1-9 and 38-41 as allegedly unpatentable over DeKoning (U.S. Pat. No. 6,691,245) and Boyd et al. (U.S. Pat. No. 6,721,806, hereinafter "Boyd").

With this Response, Applicants amend claim 2. Reconsideration is respectfully requested.

I. ART-BASED REJECTIONS

A. Claim 1

Claim 1 stands rejected over DeKoning and Boyd.

DeKoning is directed to data storage with host-initiated synchronization and fail-over of remote mirror.[1] In particular, DeKoning is directed to a system to "serv[e] the storage requirements of the client devices 104."[2] In the mirroring and fail-over system of DeKoning, what appears to be important is that the **data** on the storage devices be accessible in the event of failures.

> [T[he client devices 104 utilize the remote host device 109 and the remote storage device 110 as a fail-over storage system in the event of a failures of the local storage device 108 and/or the local host device 106.[3]

The theme regarding the data being the important aspect of DeKoning is repeated throughout.

> The business continuance client 115 then instructs the client devices 104 to switch to using the remote host device 109 and the remote storage devices 110 **for the primary data storage**.[4]

> To ensure quick and reliable fail-over to the remote host device 109 and storage device 110, the local host device 106 periodically initiates a "checkpoint," … **to synchronize *data* stored throughout the mirrored storage system 102**.[5]

---

[1] DeKoning Title.

[2] DeKoning Col. 5, lines 42-43.

[3] DeKoning Col. 5, lines 34-38.

[4] DeKoning Col. 5, lines 52-54 (emphasis added).

[5] DeKoning Col. 5, lines 58-62 (emphasis added).

Thus, there does not appear to be process (*i.e.*, a program) in DeKoning that must be provided proper state data to continue in the event of a failure, merely a continuity of data accessible by client devices 104. Stated otherwise, the "checkpoint" data in DeKoning is merely to ensure two disks hold the same data (or can be rewound to a particular point), not to ensure a backup process can pickup at the same point as a failed process.

Moreover, the communication of "checkpoint" data in DeKoning is consistent with the data continuity (rather than process continuity) theme. Dekoning's Figure 1 is reproduced for convenience of the discussion.

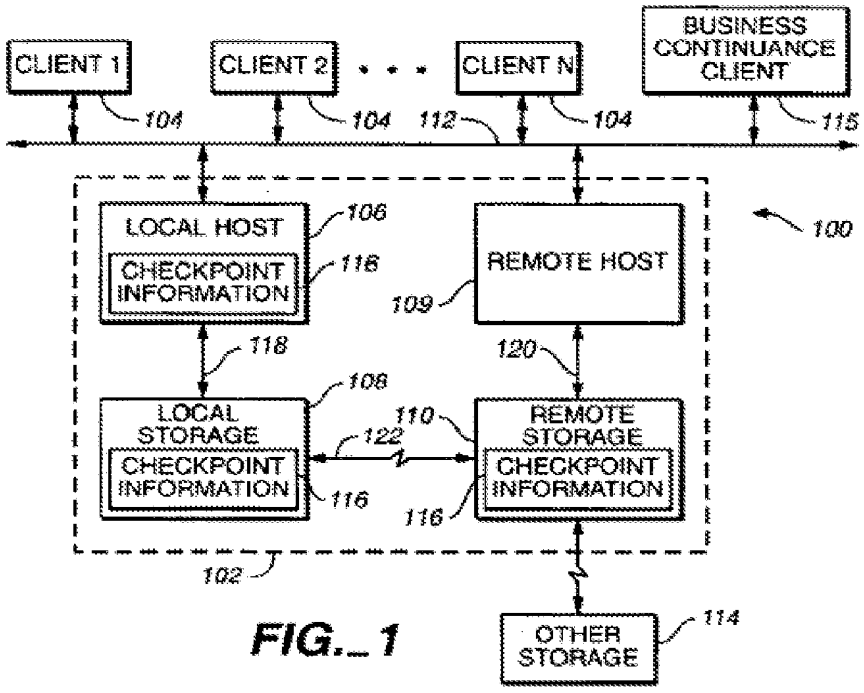

FIG._1

In particular, the flow of "checkpoint" data begins with the local host 106, and flows between the local host 106 and the local storage 108 across the link 118.

> [T]he local host 106 generates and stores the checkpoint information 116 and sends the checkpoint information 116 in a message to the local device 108.[6]

[6] DeKoning Col. 8, lines 54-57.

It is the local storage device 108 itself that sends the checkpoint data to the remote storage device 110, not the local host.

> The local storage device 108 then forwards the checkpoint information 116 in the message to the remove device 110.[7]

Thus, the flow of "checkpoint" data is from the local host 106, across the communication link 118, to the local storage 108, and then from the local storage 108, across the communication link 122, to the remote storage 110.

Claim 1, by contrast, specifically recites:

> 1. A system for storing checkpoint data comprising:
> a network interface to an external network; and
> a persistent memory unit coupled to the network interface, wherein:
> the persistent memory unit is configured to receive the checkpoint data into a region of the persistent memory unit via a remote direct memory write command from a primary process through the network interface, and to provide access to the checkpoint data in the region via a remote direct memory read command from a backup process through the network interface, wherein the remote direct memory write command is preceded by a create request for the region and the read command is preceded by an open request for the region; and
> the backup process provides recovery capability in the event of a failure of the primary process.

Applicants respectfully submit that DeKoning and Boyd fail to teach or suggest such a system. In particular, DeKoning is directed to what, in effect, is a system to ensure data is properly mirrored. DeKoning fails to address a primary process (*i.e.*, a program) and a backup process. The "checkpoint" data of DeKoning does not appear to be used by the remote host 109 process; rather, the remote host 109 is merely a conduit for access to the data by clients 104. Thus, even if

[7] DeKoning Col. 8, lines 60-62.

hypothetically the teachings of Boyd are precisely as the Office action suggests (which Applicants do not admit), DeKoning and Boyd still fail to teach or suggest "the persistent memory unit is configured to receive the checkpoint data ... via a remote direct memory write command from **a primary process** ..., and to provide access to the checkpoint data ... from **a backup process** through the network interface ... the backup process provides recovery capability in the event of a failure of the primary process." For this reason alone the rejection should be overturned and the claim set for issue.

What is more, DeKoning and Boyd fail to teach or suggest the flow of data or the access of the claims. In DeKoning, the "checkpoint" data flows in a chain from the local host 106, to the local storage 108, and then to the remote storage 110. However, in claim 1 the process checkpoint data is accessed in more of a "star" pattern, where the data is placed at a location by the primary process through a network interface, and the data is read by the backup process through the same network interface. "[T]he persistent memory unit is configured to receive the checkpoint data into a region of the persistent memory unit **via a remote direct memory write command from a primary process *through the network interface***, and **to provide access** to the checkpoint data in the region via a remote direct memory read command from a backup process **through *the* network interface**... ." Thus, even if hypothetically the teachings of Boyd are precisely as the Office action suggests (which Applicants do not admit), DeKoning and Boyd still fail to teach the placing and reading of the process checkpoint data through the same network interface.

Based on the foregoing, Applicants respectfully submit that claim 1, and all the claims which depend from claim 1 (claims 2-9 and 38-41), should be allowed.

B. Claim 2

Claim 2 stands rejected over DeKoning and Boyd. Applicants amend claim 2 to obviate the apparent interpretation by the Examiner of the "persistent memory manager configured to provide information," and not to define over any cited art. Stated otherwise, the amendment is to obviate an unreasonable interpretation of the claim language, not as an admission that the cited art taught

the prior limitations. The amendment finds support in the Specification Paragraph [0044].

Claim 2 is allowable for the same reasons as claim 1, and claim 2 also recites:

> 2. The system of Claim 1, further comprising:
> a persistent memory manager configured to program the network interface with information used by the network interface to perform virtual-to-physical address translation.

The Office action relies on DeKoning local storage device providing access to storage for the limitations of claim 2. In particular, the Office action states, "local storage device service the storage, database or other requests of the various client devices, col. 5, lines 23-26."[8] The cited location reads:

> The local host device 106 utilizes the local storage device 108 to service the storage, database or other access requests of the various client devices 104.[9]

The cited location fails to speak to a persistent memory manager, or any device for that matter, that programs the network interface with any information. Thus, even if hypothetically the teachings of Boyd are precisely as the Office action suggests (which Applicants do not admit), DeKoning and Boyd still fail to teach or suggest "a persistent memory manager configured to program the network interface with information used by the network interface to perform virtual-to-physical address translation."

Based on the foregoing, Applicants respectfully submit that claim 2 is allowable over the cited art.

II. CONCLUSION

In the course of the foregoing discussions, Applicants may have at times referred to claim limitations in shorthand fashion, or may have focused on a particular claim element. This discussion should not be interpreted to mean that

[8] Office action of December 30, 2009.

[9] DeKoning Col. 5, lines 23-26.

the other limitations can be ignored or dismissed. The claims must be viewed as a whole, and each limitation of the claims must be considered when determining the patentability of the claims. Moreover, it should be understood that there may be other distinctions between the claims and the cited art which have yet to be raised, but which may be raised in the future.

Applicants respectfully request reconsideration and that a timely Notice of Allowance be issued in this case. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's Deposit Account No. 08-2025.

Respectfully submitted,

/mes/

Mark E. Scott
PTO Reg. No. 43,100
CONLEY ROSE, P.C.
(512) 610-3410 (Phone)
(512) 610-3456 (Fax)
ATTORNEY FOR APPLICANTS

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
3404 E. Harmony Road
Fort Collins, CO 80528-9599